3D Visual Grounding with Graphs and Language Self Attention

Dominik Schmauser Tobias Drees
Technical University of Munich
{dominik.schmauser, tobias.drees}@tum.de

Abstract

In this work, we study the task of 3D object localization using referring natural language expressions. We use RGB-D scans of indoor scenes represented in the form of 3D point clouds from the recently introduced ScanRefer dataset. The corresponding model ScanRefer treats each object individually and therefore lacks context-awareness. Our key technical contribution is designing an approach leveraging a graph neural network and a language self-attention mechanism to improve the understanding of relationships between objects within a scene. We show that our model has a better understanding of the language expressions and the interactions between the objects.

1. Introduction

Object localization given a language input, also referred to as visual grounding, has benefited greatly from recent advancements in object detection and semantic language understanding. Most research has focused on a 2D setting, however for the potential applications of these techniques such as assistive robotics, an extension to a 3D context is highly desirable. A greater awareness of the interactions and relationships between the objects in a 3D scene is necessary. Various methods and new visio-linguistic datasets [1, 2, 3] have been developed, but just recently for the task of 3D visual grounding. Latest work [2] introduced a fused descriptor from 3D object proposals and sentence embeddings to accomplish state-of-the-art results. Yet, the method still suffers from a limited understanding of the interactions between the objects, in particular when multiple instances of the same object type are present, which can lead to wrong predictions. In our work, we tackle this problem by improving the extracted language features and the object relationship presentation of the ScanRefer architecture. Specifically, we add a self-attention mechanism to the language feature extraction pipeline and a graph neural network modeling local neighborhood information after the fusion module. The resulting language embeddings capture the different dependencies between the words better than solely regular recurrent architectures such as Long-Short Term memory networks and thus produce a better understanding of the scene description. Self-attention modules have been successfully applied in 2D visual grounding before [10, 4]. The graph neural network as part of the fusion of language and object features leads to object and scene-context aware output representations. We show that our method outperforms the current ScanRefer model on the ScanRefer benchmark.

2. Method

2.1. Network architecture

The fundamental architecture of the detection and encoding is identical to the original ScanRefer implementation. In addition to the x, y, z coordinates for each point, normals and multiview projections are generated to improve the quality of the object representation $N_p \in \mathcal{R}^{135}$. A PointNet++ backbone is used to transform the input point clouds into high level feature maps. A voting module [8] creates clusters with object proposals $M \in \mathcal{R}^{128}$, which are then concatenated with the language features. A proposal module outputs a bounding box proposal and an objectness mask for each cluster. The language pipeline tokenizes the input with SpaCy [5] using pretrained GloVe embeddings and a GRU. As shown in Figure 1, a language self-attention module is added on top of the GRU hidden states $N_{HS} \in \mathcal{R}^{256}$. Moreover, a graph module containing a skip connection promoting gradient flow fuses language and object features before a localization module predicts confidence scores for each object proposal. This whole pipeline is trained end-to-end.

2.2. Graph module

Our module to fuse the object and language features consists of a dynamic graph-convolutional network (DGCN) [11]. Its main component is an edge convolution, which first computes edge features of neighboring nodes

$$e'_{ijm} = ReLU(\theta_m(x_j - x_i) + \phi_m x_i) \quad (1)$$

and then aggregates all incoming edge features using the permutation invariant max operator to update the node embedding

$$x'_{im} = max\ e'_{ijm}. \quad (2)$$

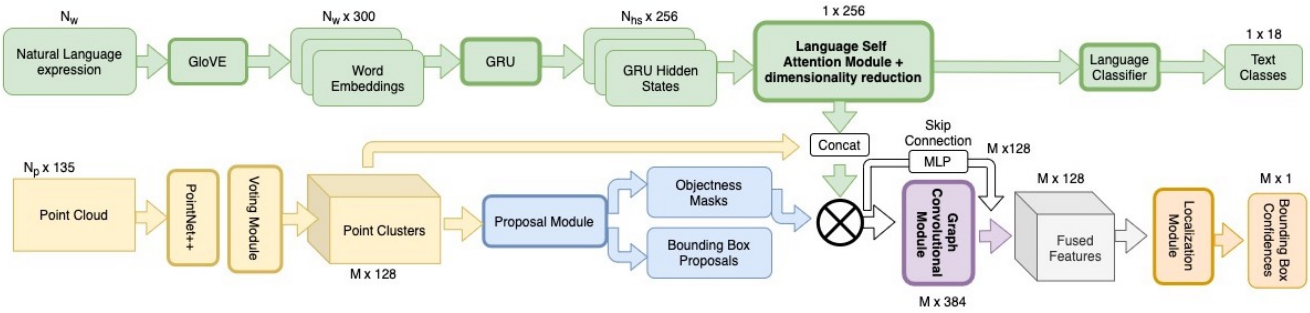

Figure 1: Architecture of ScanRefer with graph and attention: The language expressions are converted to language features from GloVE embeddings, a GRU and a language self-attention module. The input point clouds are transformed into point feature maps and a voting module produces point clusters. The suggested point clusters are concatenated with the language features and then masked with an objectness mask. A graph module with multiple edge convolutional layers is transforming the concatenated input features into an enhanced representation. The localization module then produces a confidence score for each object proposal.

This allows the network to capture interactions between objects to model local neighborhood information, as well as preserving global structure. Fusing object and language features leveraging a DGCN leads to object- and scene-context aware output representations. The initial graph is constructed by computing the k-nearest neighbors of each object feature. After each graph convolutional layer the set of k-nearest neighbors is recomputed based on the updated features resulting in a dynamic graph. This method of computing the k-nearest neighbors leads to bad interpretability, since the object features are in a transformed feature space and not related to the geometry of the original scene anymore. Hence, we additionally investigate the approach of computing the k-nearest neighbors based on the proposed object bounding box centers. This method allows for good interpretability in the graph construction step, but makes the graph static. In our experiment, we choose k = 7 for both approaches, ensuing sparse graphs with 7 nodes. This hyperparameter is chosen, since it is less computationally costly and shows promising results in recent literature [1]. We add a skip connection to the graph module to prevent the model from overfitting and ensure gradient flow. A MLP in the skip connection reduces the orginal feature dimension to match the dimension of the graph module output.

2.3. Language self-attention

In the original ScanRefer model, the input description is tokenized with pretrained GloVe word embeddings [7], which in turn are fed into a GRU. The final hidden state of the GRU is then used as the language embedding. In our adapted version, the output features for each timestep of the GRU are dotted with a set of summation weight vectors producing an interpretable sentence embedding [6]. The summation weight vectors are computed with

$$a = softmax(w_{s2} tanh(W_{s1} H^T)). \quad (3)$$

The first matrix multiplication with the weight matrix W_{s1} creates a linear combination of the hidden states of the GRU and transforms a variable length sentence into a fixed size embedding. Both matrices W_{s1} and w_{s2} are linear layers optimized through backpropagation and the softmax function normalizes the weights into output scores. This mechanism focuses on a specific semantic characteristic of the input sentence, however the sentence might contain multiple interdependencies between words. Hence, multiple "hops" of attention are computed similar to the concept of multi-head attention in the Transformer [9]. The w_{s2} vector with dimension d_a is then extended to the r-by-d_a matrix W_{s2} for r hops and summation vector a becomes summation matrix A. To ensure heterogeneity of weights across the different hops, a penalization term is introduced. The Kullback-Leibler divergence is usually used for the comparison of summation vectors, however it is unstable in this case and does not encourage each weight row to focus on a single semantic aspect. Hence, we calculate the measure of redundancy, where F represents the Frobenius norm of a matrix

$$P = \|AA^T - I\|_F^2. \quad (4)$$

This penalization term is then minimized with the overall loss. In the original paper, the final loss is a combination of the localization loss, object detection loss and the language to object classification loss

$$\mathcal{L} = \alpha \mathcal{L}_{loc} + \beta \mathcal{L}_{det} + \lambda_{scaling} \gamma \mathcal{L}_{cl}. \quad (5)$$

α, β and γ are the loss weight terms and have been originally tuned to be 0.1, 10 and 1 to achieve similar magnitude. The language to object classification loss, however,

	unique		multiple		overall	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
Baseline	76.33	**53.51**	32.73	21.11	41.19	27.40
DGCN-A 3 HL	77.41	53.17	33.38	**21.74**	41.92	**27.84**
DGCN-A 4 HL	77.33	52.38	33.20	20.71	41.76	26.86
DGCN-B 3 HL	76.94	51.70	33.69	21.31	42.09	27.21
DGCN-A static	76.12	51.47	33.51	21.32	41.78	27.17
DGCN-A 3 HL + Pretrained	**77.53**	52.51	32.33	20.92	41.10	27.05
Attention max pool	75.64	50.51	33.24	21.25	41.47	26.93
Attention linear	77.13	51.58	**34.00**	21.53	**42.37**	27.36
Test results (ScanRefer benchmark)						
Baseline	68.59	43.53	34.88	20.97	42.44	26.03
Ours (DGCN-A + Attention)	**74.94**	**51.28**	**36.31**	**22.18**	**44.97**	**28.71**

Table 1: Comparison of localization percentage results obtained by the ScanRefer with graph and attention models. Boxes whose IoU with the ground truth boxes are greater than 0.25 and 0.5 respectively are counted as correct predictions.

becomes larger due to the penalization term. Hence, we introduce a scaling term $\lambda_{scaling}$ for the loss such that the magnitude does not change.

3. Results

3.1. Quantitative results

We adopt the same metric of the tresholded intersection over union (IoU) with the ground truth to ensure comparability to the baseline model. We show that using a DGCN improves the accuracy of predicting the correct object bounding box in two common metrics, denoted IoU25 and IoU50. Especially for the setting with multiple objects of the same type in one scene, a DGCN improves performance due to its context-awareness. A hyperparameter search over the number of edge convolutional layers results in the best DGCN architecture with 3 hidden layers. The first graph method DGCN-A takes as an input the concatenated language and object features and outputs a fused feature representation. Furthermore, a modified approach DGCN-B is assessed performing the graph convolutional steps solely on the object features and fusing the language features afterwards. Building a static graph constructed on bounding box center coordinates leads to a lower accuracy in comparison to a dynamic graph. This approach produces slightly worse performance. Results are summarized in Table 1.

The main reason for the only minor improvement when using a DGCN is related to inaccurate bounding box proposals from the ScanRefer backbone. Multiple bounding box proposals for one object or proposals for non-existing objects cause weak graphs meaning many graph nodes learn from edge feature of corrupt nodes. To address this issue, we explore the use of a VoteNet pretrained on the whole ScanNet dataset, which in theory should provide better weights resulting in more accurate bounding box proposals. Our practical experiments with this setting results in a slightly worse performance in comparison to a model trained end-to-end from scratch.

We also show that self-attention has a positive influence on the localization results, mainly for the Acc@0.25 metric overall and in scenes with multiple occurances. The most impactful hyperparameter to tune in the self-attention module is the number of hops. The suggested number of 30 hops [6] also proves to be the most effective in our experiment. Furthermore, we set the penalization term to 0.5 to prevent attention weights from collapsing to a single word. The introduction of the penalization term in combination with the scaling term for the loss improves performance and is used for all experiments. We test two techniques to reduce the dimension of the output of the attention block $a_i \in \mathcal{R}^{num\_hops \times 256}$ to the original dimension of the language embedding $e \in \mathcal{R}^{256}$. A max-pool operation across the hops and a linear layer are compared. The linear layer shows better performance due to the learnable nature of the operation and is therefore selected for our final approach. The final submission on the ScanRefer test set with the combined model achieves slightly better accuracy than the baseline on both metrics.

3.2. Qualitative results

Figure 2 compares the results of the original ScanRefer model and our method. As shown in the first three examples, our model can handle scenes with multiple objects of the same type better and predicts the correct bounding box,

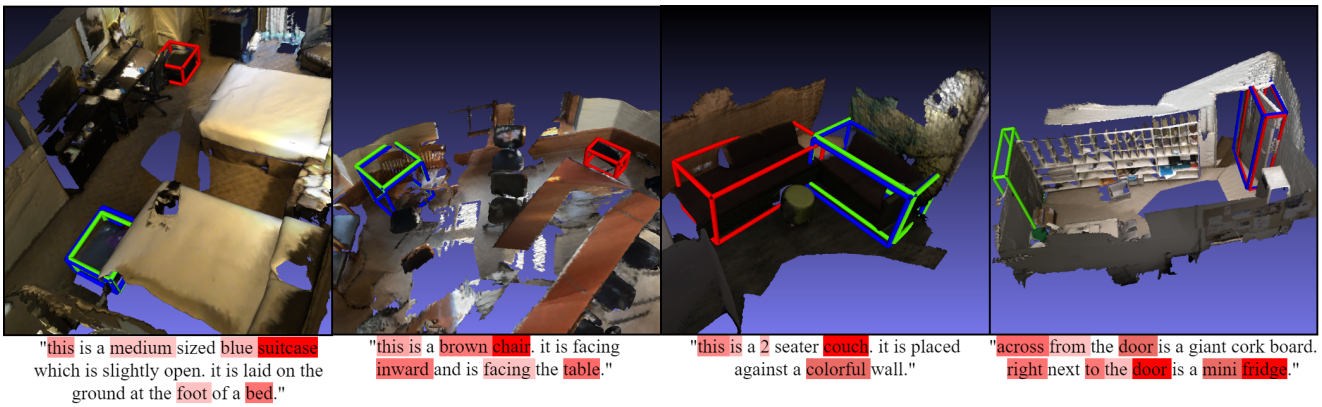

Figure 2: Qualitative results from ScanRefer baseline and our method. The blue bounding boxes represent ground truth, the red bounding boxes are the ScanRefer predictions and the green bounding boxes show our results. The strength of the red background coloring represents the size of the attention weights.

due to the scene- and context-aware nature of the graph convolutional network. The attention weights show, that the model is able to capture the relationships between the objects from the language expression and can thus choose the accurate object proposal. The attention mechanism recognizes properties about the object (such as the chair being brown and facing inward in the second example) and identifies both the relevant objects in vicinity (such as the table) and the orientation towards them (facing the table). The last picture shows a failure case, in which the attention weights focus on the relevant words, yet the model fails to infer local structure and predict the correct object bounding box.

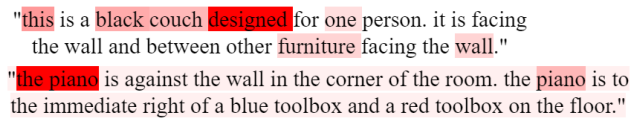

Figure 3: Failure cases of the attention mechanism

Figure 3 shows two failure cases of the attention mechanism. The failure case of the first sentence suggests that the mechanism focuses on words that are used very infrequently to describe scenes such as ”designed” and overstates its importance for the scene description. The second sentence shows an overemphasis of the attention weights on the object itself and not enough on the neighboring objects and its interdependencies.

4. Conclusion

In this work, we investigate the use of self-attention and a graph convolutional network to increase the context-awareness of the 3D visual grounding model ScanRefer. We implement a self-attention block, that enables a better understanding of the scene descriptions. We prove the valid attendance to the target object with its properties and correlated neighboring objects of each scene for most of the natural language expressions by visualising the attention weights. However, in some cases the attention mechanisms fails to attend to all relevant information. A graph neural network is applied as part of the fusion of language and object features and improves the model's understanding of the local neighborhood and global structure. The improvement in the prediction accuracy for scenes with multiple objects from one type underlines this statement. Yet, the performance of the model is still limited by the quality of the object proposals and major improvements can be achieved by enhancing the current detection pipeline of the point clouds to generate better bounding box proposals. Additionally, a broader hyperparameter search can improve the performance further, in particular for our combined DGCN and language attention approach.

Acknowledgement

We would like to thank our supervisor Dave Zhenyu Chen for the continued support on this project and the provided ScanRefer codebase.

References

[1] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. *16th European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[2] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. *16th European Conference on Computer Vision (ECCV)*, 2020. 1

[3] Dave Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X. Chang. Scan2cap: Context-aware dense captioning in rgb-d scans, 2020. 1

[4] C. Deng, Q. Wu, Q. Wu, F. Hu, F. Lyu, and M. Tan. Visual grounding via accumulated attention. In *2018 IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 7746–7755, 2018. 1

[5] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 2017. 1

[6] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130, 2017. 2, 3

[7] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. 2

[8] Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. Deep hough voting for 3d object detection in point clouds. *CoRR*, abs/1904.09664, 2019. 1

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017. 2

[10] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. *CoRR*, abs/1812.04794, 2018. 1

[11] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *CoRR*, abs/1801.07829, 2018. 1